МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

Высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВПО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Вычисление значения функции.

Отчет по лабораторной работе №3 дисциплины

«Программирование»

Выполнил студент группы ИВТб-11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Бурдукова А.А./

Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Чистяков Г.А./

Киров 2018

**Цель работы:** освоить синтаксис построения процедур и функций, изучить способы передачи данных в подпрограммы, получить навыки организации минимального пользовательского интерфейса

**Задание:**

**1. Реализовать программу вычисления площади фигуры, ограниченной кривой**

**-5\*x^3+(43)\*x^2+(274)\*x+(-1026).**

**2. Вычисление определенного интеграла должно выполняться численно, с применением метода средних прямоугольников.**

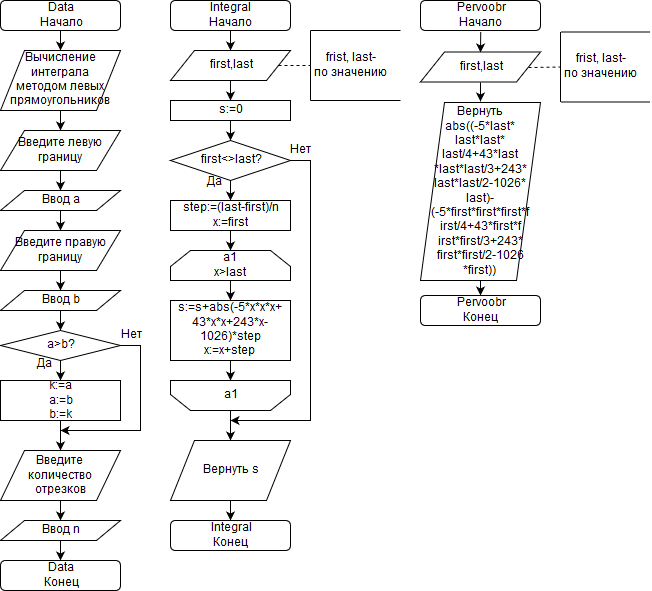
**3. Пределы интегрирования вводятся пользователем.**

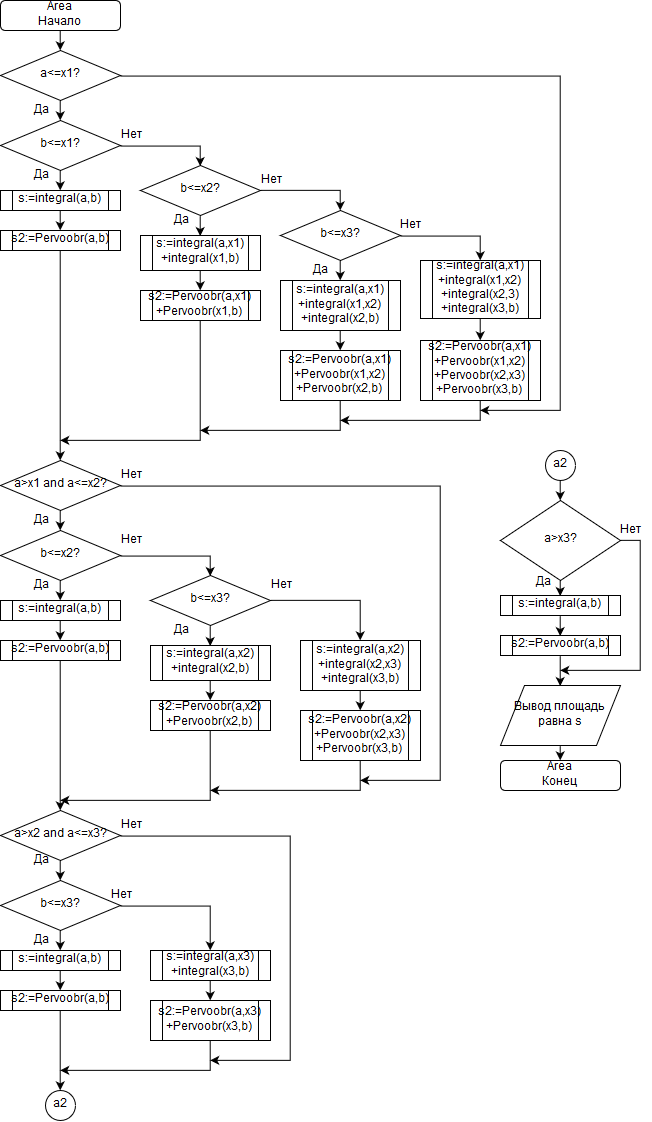
**4. Взаимодействие с пользователем должно осуществляться посредством case-меню.**

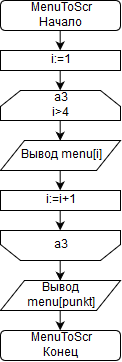
**5. Требуется реализовать возможность оценки погрешности полученного результата.**

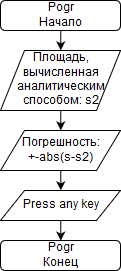
**6. Необходимо использовать процедуры и функции, где это целесообразно**

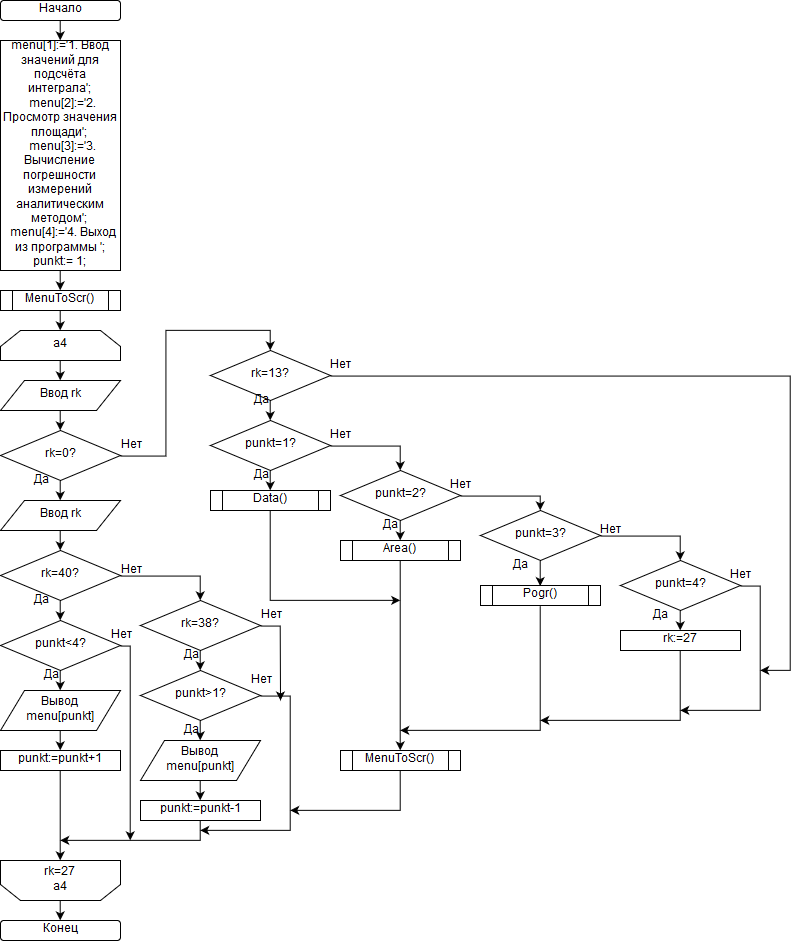
**Схема алгоритмов:**

****

****

****

****

****

**Исходный текст программы:**

**uses** crt;

**const** x1=-6.053; x2=2.879; x3=11.774;

**var** a,b,step,s,s2,k :real;

punkt: byte;

n : longint;

rk : char;

menu : **array**[1..4] **of** string;

buf : string;

err : integer;

**procedure** Data();

**begin**

clrscr;

writeln ('Вычисление интеграла методом левых прямоугольников');

writeln();

write('Введите левую границу: ');

readln(buf);

val(buf, a, err);

**while** (err <> 0) **or** (length(buf) > 3) **do**

**begin**

writeln('Левая граница может быть только натуральным числом, не првеосходящим 100');

write('Введите левую границу: ');

readln(buf);

val(buf, a, err);

**end**;

write('Введите правую границу: ');

readln(buf);

val(buf, b, err);

**while** (err <> 0) **or** (length(buf) > 3) **do**

**begin**

writeln('Правая граница может быть только натуральным числом, не првеосходящим 100');

write('Введите правую границу: ');

readln(buf);

val(buf, b, err);

**end**;

**if** (a > b) **then**

**begin**

k:=a;

a:=b;

b:=k;

**end**;

write('Введите количество отрезков (только натуральное число, меньшее 1 000 000): ');

readln(buf);

val(buf, n, err);

**while** (err <> 0) **do**

**begin**

write('Количество отрезков может быть только натуральным числом, повторите ввод: ');

readln(buf);

val(buf, n, err);

**end**;

**end**;

**function** Integral(first,last:real):real;

**var** x:real;

**begin**

s:=0;

**if** (first <> last) **then**

**begin**

step:=(last-first)/n;

x:=first;

**while** (x<=last) **do**

**begin**

s:=s+abs(-5\*x\*x\*x+43\*x\*x+243\*x-1026)\*step;

x:=x+step;

**end**;

**end**;

Integral:=s;

**end**;

**function** Pervoobr(first,last:real):real;

**begin**

Pervoobr := abs((-5\*last\*last\*last\*last/4+43\*last\*last\*last/3+243\*last\*last/2-1026\*last)-(-5\*first\*first\*first\*first/4+43\*first\*first\*first/3+243\*first\*first/2-1026\*first));

**end**;

**procedure** Area();

**begin**

**if** (a<=x1) **then**

**begin**

**if** (b<=x1) **then**

**begin**

s:=Integral(a,b);

s2:=Pervoobr(a,b);

**end**

**else**

**if** (b<=x2) **then**

**begin**

s:=Integral(a,x1)+Integral(x1,b);

s2:=Pervoobr(a,x1)+Pervoobr(x1,b);

**end**

**else**

**if** (b<=x3) **then**

**begin**

s:=Integral(a,x1)+Integral(x1,x2)+Integral(x2,b);

s2:=Pervoobr(a,x1)+Pervoobr(x1,x2)+Pervoobr(x2,b);

**end**

**else**

**begin**

s:=Integral(a,x1)+Integral(x1,x2)+Integral(x2,x3)+Integral(x3,b);

s2:=Pervoobr(a,x1)+Pervoobr(x1,x2)+Pervoobr(x2,x3)+Pervoobr(x3,b);

**end**;

**end**;

**if** (a>x1) **and** (a<=x2) **then**

**begin**

**if** (b<=x2) **then**

**begin**

s:=Integral(a,b);

s2:=Pervoobr(a,b);

**end**

**else**

**if** (b<=x3) **then**

**begin**

s:=Integral(a,x2)+Integral(x2,b);

s2:=Pervoobr(a,x2)+Pervoobr(x2,b);

**end**

**else**

**begin**

s:=Integral(a,x2)+Integral(x2,x3)+Integral(x3,b);

s2:=Pervoobr(a,x2)+Pervoobr(x2,x3)+Pervoobr(x3,b);

**end**;

**end**;

**if** (a>x2) **and** (a<=x3) **then**

**begin**

**if** (b<=x3) **then**

**begin**

s:=Integral(a,b);

s2:=Pervoobr(a,b);

**end**

**else**

**begin**

s:=Integral(a,x3)+Integral(x3,b);

s2:=Pervoobr(a,x3)+Pervoobr(x3,b);

**end**;

**end**;

**if** (a>x3) **then**

**begin**

s:=Integral(a,b);

s2:=Pervoobr(a,b);

**end**;

clrscr();

writeln('Площадь равна: ',s:0:5);

write('Press any key');

Readkey();

**end**;

**procedure** Pogr();

**begin**

clrscr;

writeln('Площадь, вычисленная аналитическим методом: ',s2);

writeln('Погрешность: +-', abs(s-s2):0:5,' ');

write('Press any key');

rk:=Readkey();

**end**;

**procedure** MenuToScr;

**var** i:integer;

**begin**

clrscr;

**for** i:=1 **to** 4 **do**

**begin**

gotoxy(1,i);

writeln(menu[i]);

**end**;

textcolor(4);

gotoxy(1,punkt);

write(menu[punkt]);

textcolor(7);

**end**;

**begin**

clrscr;

textcolor(7);

writeln ('Функция для подсчёта площади методом левых прямоугольников такова: ');

writeln('F:=-5\*x^3+43\*x^2+274\*x-1026');

writeln('Корни данного уравнения: ',x1,' ',x2,' ',x3);

writeln ('Используйте стрелки вверх и вниз для перемещения по пунктам меню и клавишу "Enter" для выбора соответствующего пункта.');

writeln ('Press any key');

Readkey();

textcolor(7);

menu[1]:='1. Ввод значений для подсчёта интеграла';

menu[2]:='2. Просмотр значения площади';

menu[3]:='3. Вычисление погрешности измерений аналитическим методом';

menu[4]:='4. Выход из программы ';

Readkey();

punkt:= 1;

MenuToScr();

**repeat**

rk := Readkey();

**if** (rk = #0) **then**

**begin**

rk := Readkey;

**case** rk **of**

#40:

**if** (punkt < 4) **then**

**begin**

gotoxy(1,punkt);

write(menu[punkt]);

inc(punkt);

Textcolor(4);

gotoxy(1,punkt);

write(menu[punkt]);

Textcolor(7);

**end**;

#38:

**if** (punkt > 1) **then**

**begin**

gotoxy(1,punkt);

write(menu[punkt]);

dec(punkt);

Textcolor(4);

gotoxy(1,punkt);

write(menu[punkt]);

Textcolor(7);

**end**;

**end**;

**end**

**else**

**if** (rk = #13) **then**

**begin**

**case** punkt **of**

1:

Data();

2:

Area();

3:

Pogr();

4:

rk:=#27;

**end**;

MenuToScr();

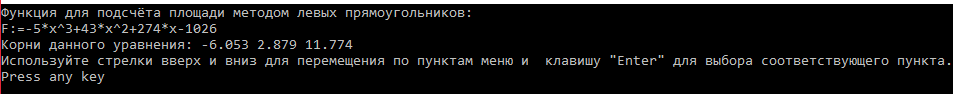
**end**

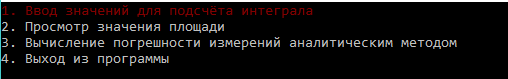
**until** rk=#27;

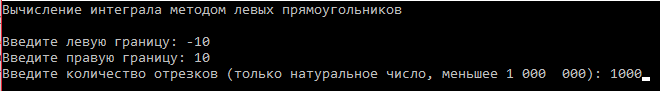
clrscr();

**end**.

**Скриншоты выполнения программы**

****

****

****

****

****

**Вывод**

В ходе данной лабораторной работы были получены знания для создания собственных подпрограмм, а именно, процедур и функций. Были освоены принципы работы в данных подпрограммах. Также удалось познакомиться с так называемым “общением” между разработчиком и пользователем при помощи case-меню. В самой программе часто использовалась такая функция, как ReadKey(), позволяющая считывать символ клавиши, нажатой на клавиатуре пользователем. Было применено перемещение курсора в консоли при помощи процедуры GoToXY. Также было осуществлено изменение цвета выводимого текста при помощи процедуры TextColor.

Лабораторная работа выполнялась в бесплатной среде разработки PascalABC. Данная среда предоставляет много возможностей, в числе которых есть отладчик, который применялся в процессе написания программы для её отладки. Таким образом, был получен навык работы с отладчиком и поиск ошибок в программе.